

## **REMARKS**

Claims 21-67 remain pending in the application. Reconsideration is respectfully requested in light of the following remarks.

### **Section 103(a) Rejections:**

The Examiner rejected claims 21, 22, 24, 25, 29-36, 38, 39, 43-45, 47-51, 53, 54, 58-65, 67 and 68 under 35 U.S.C. § 103(a) as being unpatentable over Cook et al. (U.S. Patent 5,727,950) (hereinafter “Cook”) in view of Martinez (U.S. Patent 5,546,521). Applicant respectfully traverses this rejection for at least the reasons presented below.

**Regarding claim 21, Cook in view of Martinez fails to teach or suggest maintaining a cache specific to help data for one or more user interface components.** Cook teaches a method for interactive, adaptive, and individualized computer-assisted instruction that provides a virtual agent or tutor adapted to each student (Cook, column 5, lines 12-24). Cook’s virtual agent presents educational material and is responsive to the student actions and requests (Cook, column 12, lines 46-51, and column 13, lines 8-27). While Cook does teach that the virtual agent provides help data to the student, Cook teaches that the system is advantageously organized *to wait for student* and timer input events and then to respond appropriately to those events (Cook, column 37, line 66 – column 38, line 2).

Contrary to the Examiner’s assertion, Cook clearly fails to teach or suggest maintaining a cache *specific to help data for one or more user interface components*. In contrast, Cook teaches that the files or lessons for computer-aided instruction may be downloaded ahead of time. However, these files are not help data for one or more user interface components, but are “educational content such as instructional units, homework assignments, and testing” materials (Cook, column 10, lines 59-64). Further, the files downloaded by Cook are not maintained in a cache specific to help data. Cook clearly

fails to teach or suggest maintaining such a cache specific to help data for one or more user interface components.

The Examiner cites Cook's teaching regarding how "[l]arge files can be downloaded in advance of a student session or the student client can cache read-only data across sessions obviating the need for downloading such files" (Cook, column 16, lines 20-22). However, Cook is clearly referring to the student's client software downloading lesson files and other educational materials for presentation to the student. Cook does not describe the downloaded files as being *help data for user interface components*. Nor does Cook mention anything regarding a cache *specific to* such help data. The files downloaded by Cook are not presented to the user as help data for user interface components. Instead the download files include "computer-assisted instruction materials which present to students a variety of interactive, adaptive, and self-paced computer-assisted instruction and homework materials in a manner which informs the agent of a student's progress and performance and which permits the agent to manage or control the materials to the student's pedagogic characteristics" (Cook, column 6, lines 57-64). Such files *are not help data for one or more user interface components*. In regard to *help data*, Cook only teaches that its virtual agent provides help data to the student, and that the system is advantageously organized *to wait for student* and timer input events and then to respond appropriately to those events (Cook, column 37, line 66 – column 38, line 2). Thus, Cook fails to teach anything regarding maintaining a cache specific to help data for one or more user interface components.

In response to the above arguments, the Examiner, in the Response to Arguments, again cites Cook's teachings regarding the downloading of large files and asserts that the files downloaded are of reading, mathematics and related topics, spelling, writing, and other language arts and cites column 3, lines 26-27. However, the cited passage comes from Cook's background section where Cook outlines problems with previous solutions to computer-based learning. The subjects listed (reading, mathematics and related topics, spelling, writing, and other language arts) are not describing Cook's system, but instead are a part of a separate system (the integrated learning system, or ILS). Cook does not

describe the large files his system downloads as including such informational topics. The Examiner has improperly attributed features of a separate prior art solution, which Cook uses to illustrate problems with previous solutions to computer based instruction, to elements of Cook's system. Cook does not suggest using any aspects of this other prior art system. In fact, if anything, Cook teaches away from employing features of prior systems such as described at the section cited by the Examiner. Furthermore, regardless of whether or not the large files downloaded in Cook include the subjects listed by the Examiner, such files are not described as *help data for one or more user interface components* and Cook's system does not include maintaining a *cache specific to help data* for user interface elements.

Additionally, the Examiner argues that the files downloaded in Cook may be cached, such as on cache disk 209. Applicant is not arguing that Cook fails to teach caching at all. Applicant's argument is that Cook fails to teach or suggest "maintaining a cache *specific to help data for one or more user interface components*." The large files downloaded in Cook are not help data for user interface components. Whether or not Cook teaches that large student lesson files can be downloaded and even cached is completely irrelevant to whether or not Cook teaches maintaining a cache specific to help data for user interface components. The Examiner is apparently arguing that teaching any sort of caching implies maintain a cache specific to help data for one or more user interface components. However, downloading and even caching other files does not imply or suggest maintain a cache specific to help data for user interface components.

The Examiner asserts that Cook teaches that "files of reading, mathematics, and related topics, spelling, writing, and other language arts can be downloaded in advance of a student session [*sic*]". Applicant submits that this issue has been addressed extensively above and in previous responses; the Examiner has not provided new arguments or rebuttals and continues to repeat the same erroneous assertions.

Cook fails to disclose anything regarding providing help for user interface components. In contrast, Cook teaches that the virtual agent may provide "meta-

cognitive help on ongoing instruction” (Cook, column 5, lines 47-49). For example, the student may make meta-requests to the virtual agent, such as “How am I doing?” to which the agent responds (Cook, column 13, lines 15-24). Cook is teaching that the virtual agent can provide educational help relating to a student’s lessons in subjects such as math or writing. This type of help, as described by Cook, is very different from help data *for user interface components*. Nowhere does Cook discuss anything regarding a help system that provides help data for one or more user interface components.

In response to the above argument, the Examiner again asserts that Cook teaches, “help data files of reading, mathematics and related topics, spellings, writing, and other language arts.” The Examiner is again referring to Cook’s teachings regarding downloading and caching large files. However, as noted above, the Examiner has erroneously and improperly attributed the subjects of a separate, prior art solution for computer based learning (ILS) to Cook’s large file downloads. Now the Examiner is also stating that such files are help data files for these subjects. The Examiner is clearly incorrect. Firstly, the subjects listed by the Examiner are not even described as files, and certainly not as help files for user interface components. Column 3, lines 23 – 30 of Cook clearly state that the integrated learning system “is used in schools to teach basic strands of reading, mathematics and related topics, spelling, writing, and other language arts, from grades one to six.” Nothing is mentioned about these subjects including help data files. Secondly, Cook does not mention that the large files referred to by the Examiner include the subjects included in the ILS system and listed by the Examiner. Thirdly, the Examiner is merely speculating that the large files downloaded in Cook’s system are in fact help data files. Finally, the subjects and files in Cook have nothing to do with user interface components.

Further in regard to claim 21, Cook in view of Martinez clearly fails to teach or suggest that in response to receiving a request for help data for a newly referenced one of the user interface components, if the help data for the referenced user interface component is not in the cache, loading the help data for the referenced user interface component into the cache and supplying the help data for the referenced user interface

component for user presentation. The Examiner admits that Cook fails to teach loading the help data for a referenced component into the cache if the help data for the referenced component is not in the cache. The Examiner relies upon Martinez for this feature. However, contrary to the Examiner's assertion, Martinez clearly fails to teach such a feature. Martinez teaches a help system including dynamically displaying help text for a user interface object under the mouse, but fails to teach anything regarding a *cache specific to help data* and additionally fails to teach loading the help data for a referenced component into the cache if the help data for the referenced component is not in the cache. In contrast, Martinez teaches that help data is loaded either from dynamic table 41 or static table 40.

Specifically, Martinez teaches that in response to determining that an object is under the mouse pointer, the help system first looks in dynamic table 41 and if it doesn't find any help text in dynamic table 41 for the object, the help system looks in static table 40 "to determine whether a help string should be displayed to the user" (Martinez, column 6, line 64 – column 7, line 2). Additionally, Martinez teaches that certain applications, that he calls aware applications, may update the help data in dynamic table 41 (Martinez, column 5, lines 49-55). However, this updating of dynamic table 41 is proactive, and *not reactive to not finding help data for a referenced user interface components*. In other words, an aware application may modify the specific help text displayed for a particular user interface element, but Martinez does not teach that help data is loaded into dynamic table 41 if no help data for a user interface component is found in dynamic table 41. Instead, as noted above, Martinez teaches looking in static table 40 and displaying any help data found, but Martinez does not teach loading any help data from static table 40 into dynamic table 41. Thus, neither dynamic table 41, nor static table 40 are caches as understood in the art, but instead are merely string tables from which help data is read and displayed to the user.

In response to the above argument, in the Response to Arguments section of the Office Action the Examiner again cites column 5, lines 53-57 of Martinez and repeats the same argument from the rejection of claim 21. Thus, the Examiner merely repeats the

same rejection argument without actually providing any actual rebuttal to Applicant's argument above. As noted above, the cited passage of Martinez only describes how Martinez' system provides dynamic help information, "such as the number and type of mail items contained in an in-basket icon" (Martinez, column 5, lines 56-60). Thus, the cited passage is describing loading of dynamic help information in addition to static help information, but does not mention anything regarding loading the help data for a referenced component into the cache if the help data for the referenced component is not in the cache. The cited passage is not loading help data for a referenced component into a cache. Instead, as described above, the cited passage is describing the ability of "aware applications" to "dynamically update the shared memory in dynamic table database 41, from which Infomouse 35 can read help information." The cited passage does not even mention anything about a referenced user interface component. Applicant fails to see the relevance of the Examiner's cited passage. **Applicant submits that this issue has been addressed extensively above and in previous responses. The Examiner has not provided new arguments or rebuttals and continues to repeat the same erroneous assertions.**

The Examiner also cites various passages from Martinez that describe different functions of Martinez' help system including: referring to a table correlating user interface objects with help data in response to determining that the pointer is over a user interface object (Martinez, column 2, lines 49-54), presenting help data from the help systems even if the underlying application does not provide its own help data (Martinez, column 6, lines 1-9), that dynamic help table 41 resides in a shared memory segment separate from static help table 40 (Martinez, column 5, lines 54-56), and describing an control application, OverSee 37, that conforms to the Common User Access guidelines for its graphical user interface (Martinez, column 6, lines 42-44). None of these sections mentions anything regarding a cache specific to help data, nor about loading into a cache help data for a user interface component for which no help data is currently in the cache.

In the Response to Arguments section, the Examiner merely repeats the same irrelevant citations from Cook (column 3, lines 26-27 and column 16, lines 20-26) and

Martinez (FIG. 2 and column 5, lines 55-57). As discussed above, the cited passages clearly fail to teach or suggest the features and functionality the Examiner attributes to them.

Further, Martinez teaches that when the mouse pointer is over an object for which the help system does not already have help information, the help system attempts to find the information associated with the object in help tables 40 and 41. If the help system cannot find help data for the object, the help system “returns to monitoring the current screen location of the mouse pointer.” (Martinez, column 8, lines 49-63). Thus, Martinez fails to teach maintaining a cache and also fails to teach loading help data for the referenced component into the cache if the help data for a referenced component is not in the cache. In contrast, Martinez explicitly teaches that if no help information is found in dynamic table 41, help text may be loaded from static table 40 and displayed to the user, but it is not loaded into any kind of cache. Nor is it loaded from static table 40 to dynamic table 41. Additionally, if no help information is found in static table 40, the help system does not load or display any help text for the referenced object but simply returns to monitoring the current screen location of the mouse pointer.

In response to the above argument, the Examiner again cites FIG. 2 and column 5, lines 55 – 57 of Martinez, without providing any actual rebuttal to Applicant’s argument. The portions of Martinez cited by the Examiner have no relevance to Applicant’s argument. The Examiner has not provided new arguments or rebuttals and continues to repeat the same erroneous assertions.

Applicant submits that, contrary to the Examiner’s assertion, neither Cook nor Martinez, either alone or in combination, teach or suggest: maintaining a cache specific to help data for one or more user interface components; and in response to receiving a request for help data for a newly referenced one of the user interface components: if the help data for the referenced component is not in the cache, loading the help data for the referenced user interface component into the cache. Furthermore, as explained above, any system resulting from the Examiner’s proposed combination of Cook and Martinez

would not include maintaining a cache specific to help data for one or more user interface components, nor would such a proposed system include loading the help data for a referenced user interface component into the cache if the help data for a referenced user interface component is not in the cache.

For at least the reasons presented above, the rejection of claim 21 is clearly not supported by the prior art and removal thereof is respectfully requested. Similar arguments apply in regard to claims 35, 49 and 63.

Regarding claim 24, Cook in view of Martinez fails to teach loading into the cache help data for one or more non-referenced user interface components *associated with a referenced user interface component*. The Examiner cites column 3, lines 26-27 and column 16, lines 20-27 of Cook and argues that Cook teaches “the technique of large files of reading mathematics **and related topics**, spellings (sic), writing, and other language arts can be downloaded in advance of a student session or the student client can cache read-only data across [sessions] obviating the need for downloading such files” (bolding by Examiner). However, as discussed above regarding claim 21, the Examiner is improperly combining features of two different systems, namely Cook’s own system and a separate prior art system called the integrated learning system (ILS) described in Cook’s background section the disadvantages of which Cook is seeking to overcome. Furthermore, the Examiner’s interpretation of Cook’s system and ILS still does not anything at all to do with loading help data for a non-referenced user interface component associated with a referenced user interface component. Not only do the cited passages fail to mention anything regarding a non-referenced user interface component associated with a referenced user interface component, the cited passages do not even mention help data or a cache specific to help data for user interface components.

The Examiner’s proposed combination of Cook and Martinez clearly would not result in a system that would include loading into the cache help data for one or more non-referenced user interface components associated with a referenced user interface



component. Thus, the rejection of claim 24 is not supported by the teachings of the cited art and withdrawal thereof is respectfully requested. Similar arguments apply in regard to claims 29, 38, 43, 53, 58 and 67.

Regarding claim 32, Cook in view of Martinez does not teach or suggest in response to receiving a preload request indicating one or more of the user interface components, loading into the cache help data for each of the indicated user interface components. The Examiner admits that Cook fails to teach this limitation and relies upon Martinez. However, Martinez fails to teach anything regarding receiving a preload request indicating one or more user interface components. Nor does Martinez teach anything about loading, into a cache, help data for each of the indicated user interface components in response to receiving such a preload request. The Examiner's cited passage (Martinez, column 5, lines 39-44 and FIG. 3) refers only to the fact that Martinez' preferred operating system is IBM's OS/2 and that his help facility 35 works with three different types of applications: aware applications capable of dynamically modifying displayed help data, applications that only display static help data, and applications that do not provide any help data. Nowhere, including in the cited passage, does Martinez (or Cook) mention anything about receiving a preload request indicating one or more user interface components nor about in response to receiving a preload request indicating one or more of the user interface components, loading into the cache help data for each of the indicated user interface components.

In contrast, as described above regarding claim 21, Martinez teaches a help system that include two string tables, dynamic table 41 and static table 40, which contain help data to be displayed when user interface elements are referenced. However, no additional help data is loaded into either of these tables in response to a preload request indicating user interface components. Thus, the proposed combination of Cook and Martinez would also fail to include receiving a preload request indicating one or more components and would also clearly fail to include in response to receiving a preload

request indicating one or more of the user interface components, loading into the cache help data for each of the indicated user interface components.

Thus, the rejection of claim 32 is not supported by the teachings of the cited art and withdrawal thereof is respectfully requested. Similar arguments apply in regard to claims 47 and 61.

The Examiner rejected claims 23, 37, 52 and 66 under 35 U.S.C. § 103(a) as being unpatentable over Cook in view of Martinez in further view of Monsen et al. (U.S. Patent 6,606,628) (hereinafter “Monsen”). Applicant traverses this rejection for at least the reasons given below.

Regarding claim 23, Cook in view of Martinez in further view of Monsen fails to teach or suggest deleting the least recently requested help data in the cache if there is not enough free space in the cache to store the help data for the referenced user interface component. The Examiner admits that Cook in view of Martinez fails to teach or suggest deleting the least recently requested help data in the cache if there is not enough free space in the cache to store the help data for the referenced user interface component. The Examiner relies upon Monsen to suggest this feature. Monsen teaches a flat (no directories) file system for systems with only a small amount of nonvolatile memory space available. The Examiner cites three passages from Monsen and argues that combined they teach deleting the least recently requested help data in the cache if there is not enough free space in the cache to store the help data for the referenced user interface component. The Examiner’s interpretation of Monsen is incorrect. The first cited passage (Monsen, column 5, line 26) is part of a large passage describing an information structure Monsen uses for each open file. This information structure includes a “delete on close” flag indicating whether the file should be deleted when it is closed. The second cited passage of Monsen (column 4, line 28) describes how in Monsen’s file system, no list of free blocks is maintained because “there are a relatively small number of blocks ... and write operations to the nvram medium are not performed very frequently.” The third

cited passage (Monsen, column 5, line 50) describes another data structure used in Monsen's file system to store the geometry of the nvram medium. Monsen teaches that sometimes an initial portion of the nvram medium may be allocated for to a different file system than Monsen's. Monsen further states, "the size offset structure member may be used to indicate the size of the nvram medium memory space allocated for other purposes" (Monsen, column 5, lines 48-50).

The Examiner combines these three cited passages described above and argues that "Monsen discloses the limitation of deleting the least recently requested help data in the cache if there is not enough free space in the cache to store the help data for the referenced component as the technique of deleting the file (see col. 5, line 26) which is not performed very frequently (see col. 4, line 28) in order to allocate memory space for other purposes (see col. 5, line 50)" (parentheses by Examiner). The Examiner has clearly misinterpreted the teachings of Monsen. Furthermore, the Examiner has clearly taken three disparate sentences from Monsen, each out of context, and combined them in an improper attempt to reconstruct the limitations of claim 23 in hindsight. The Examiner has changed the meaning of each of the three cited sentences, which, even when taken together, have nothing whatsoever to do with help data. The cited passage certainly do not teach or suggest deleting the least recently requested help data in the cache if there is not enough free space in the cache to store the help data for the referenced component. Not only does Monsen not make any mention of help data, the three cited passages (even when erroneously combined as suggested by the Examiner) do not describe any sort of caching. Instead, each passage describes a different aspect of Monsen's file system.

Additionally, the Examiner's interpretation of the three cited passages does not teach or suggest the functionality the Examiner argues. Specifically, the Examiner (incorrectly) interprets the combination of the three cited passages as "deleting the file which is not performed very frequently in order to allocate memory space for other purposes." However, this interpretation does not teach or suggest deleting a least recently requested help data in a cache if there is not enough free space in the cache to

store help data for a referenced component. The Examiner's interpretation of Monsen's teachings has nothing to do with help data or a help data cache or referenced user interface components. Monsen clearly fails to overcome the deficiencies of Cook and Martinez regarding deleting the least recently requested help data in a cache if there is not enough free space in the cache to store help data for a referenced component.

The suggested combination of Cook, Martinez and Monsen would not result in a system that included deleting a least recently requested help data in a cache if there is not enough free space in the cache to store help data for a referenced user interface component. Instead, the Examiner's proposed combination of Cook, Martinez and Monsen would result in a system of computer based training as taught by Cook, which displayed contextual help or status information as taught by Martinez and that used the NVRAM file system of Monsen.

Thus, Cook, Martinez, and Monsen, whether considered separately or in combination, fail to teach or suggest deleting the least recently requested help data in the cache if there is not enough free space in the cache to store the help data for the referenced component. For at least the reasons presented above, the rejection of claim 23 is not supported by the teachings of the cited art and removal thereof is respectfully requested. Similar arguments apply in regard to claims 37, 52 and 66.

The Examiner also rejected claims 26-28, 40-42 and 55-57 under 35 U.S.C. § 103(a) as being unpatentable over Cook in view of Martinez in further view of Medl et al. (U.S. Patent 6,209,006) (hereinafter "Medl"). Applicant traverses this rejection for at least the reasons presented below.

Regarding claim 26, Cook in view of Martinez in further view of Medl fails to teach or suggest loading into the cache additional help data indicated by one or more hyperlinks in the help data for the referenced user interface component. The Examiner relies upon Medl, citing column 2, lines 48-56. Medl teaches a system in which

hyperlinks in the content of applications are used to provide help data. However, the hyperlinks in Medl are in an application's content, so that a user can retrieve help information "without having to stray from the interface of the application program itself" (Medl, column 2, lines 63 – 66). Thus, Medl teaches loading help information from hyperlinks in applications. Medl does not mention anything regarding hyperlinks *in the help data for a referenced component*. In fact, Medl does not discuss the content of help data at all and certainly does not describe loading into a cache help data indicated by hyperlinks in the help data for a referenced user interface component.

In the response to arguments, the Examiner asserts the motivation to combine Martinez and Cook: "the system would be enhanced by capable of tracking what kind of help its end user needed, and updating from dynamic table storage, and presenting help information to its end user. Thus, the system would provide an enhanced contextual help to its end user". Applicant submits that this is not a proper motivation and notes that neither Cook nor Martinez disclose the claimed features, and further, neither indicates or even hints at the desirability of these features. Thus, Applicant submits that the attempted combination of these references, as argued extensively above, would not produce Applicant's invention as represented in claim 26.

Since Cook, Martinez and Medl all fail to teach or suggest loading into the cache additional help data indicated by one or more hyperlinks in the help data for the referenced user interface component, the Examiner's proposed combination of Cook, Martinez and Medl would also fail to teach such functionality. Thus, the rejection of claim 26 is not supported by the cited prior art and removal thereof is respectfully requested. Similar remarks apply to claims 27, 28, 40-42 and 54-56.

Applicant also asserts that the rejections of numerous ones of the dependent claims are further unsupported by the cited art. However, since the rejections have been shown to be unsupported for the independent claims, a further discussion of the dependent claims is not necessary at this time.

## CONCLUSION

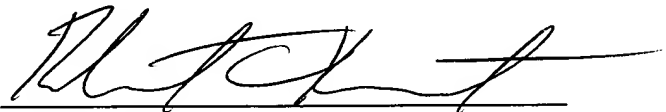
Applicant submits the application is in condition for allowance, and prompt notice to that effect is respectfully requested.

If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5681-85700/RCK.

Also enclosed herewith are the following items:

☒ Return Receipt Postcard

Respectfully submitted,



Robert C. Kowert  
Reg. No. 39,255  
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.  
P.O. Box 398  
Austin, TX 78767-0398  
Phone: (512) 853-8850

Date: November 18, 2005